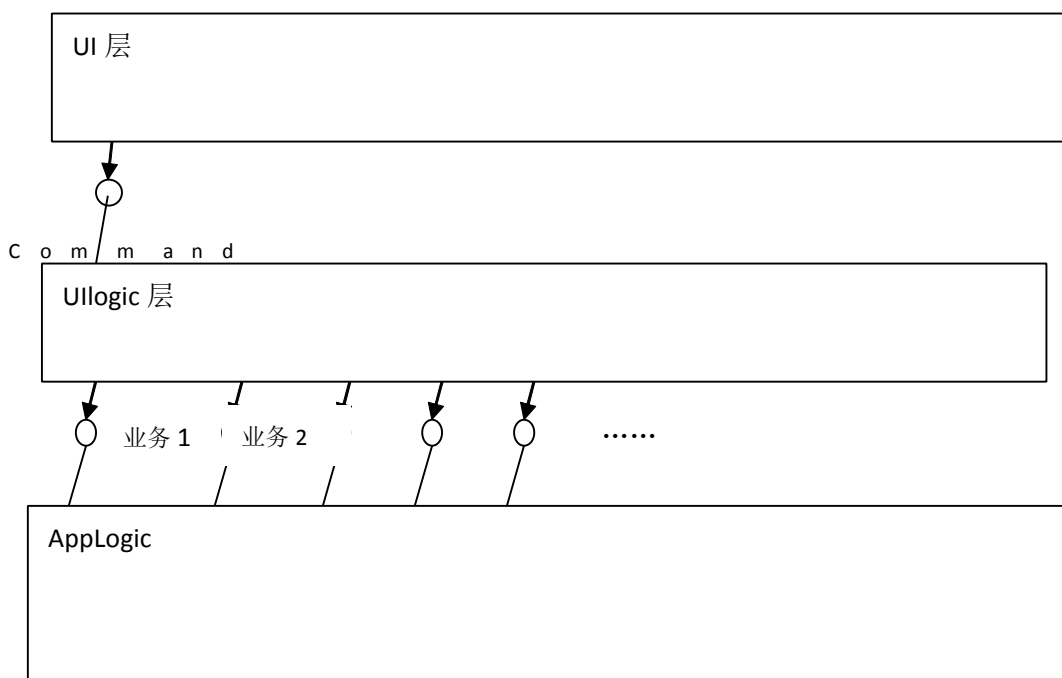


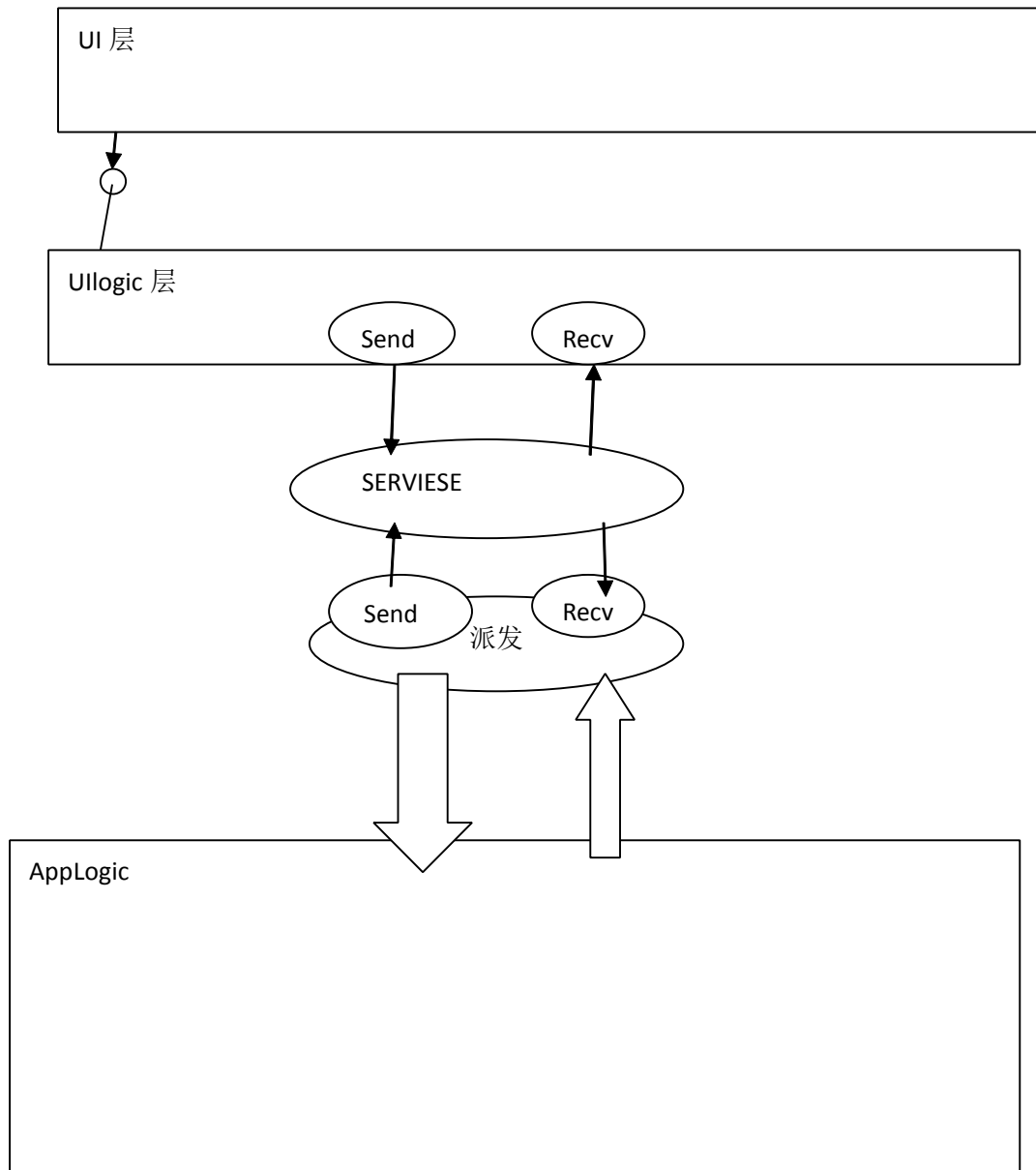
<http://www.LomoX.hk> 社区发布。

传统的界面 `ui` 和 `uilogic` 隔离方式采用的是接口的方式，为了解耦合，为了使隔离层之间业务统一。如图，`ui` 层你就负责界面的，你想用什么语言都可以；`uilogic` 层你就只做 `ui` 传下来的 `io`，键盘鼠标等的操作行为解析；`applogic` 你就负责业务，计算执行的任务。这样子，一个需求就可以同时三个人开发了。但是有个问题，对接的时候到处都是接口，这个一定要提前定义好，否则在对接的时候就麻烦了。用了好几年你会发现到处都是接口，业务的抽象接口居多，其次是统一抽象的 `command` 的接口。对于业务接口的膨胀，那可是一个无语啊，出现问题，定位一个问题 你一定要知道这玩意跑的是啥接口，这个业务写的接口名叫啥。



大型基础软件为了解耦而做的隔离，导致接口声明膨胀。可以看到 `applogic` 层和 `uilogic` 的接口会无限膨胀，与其带来的就是维护成本，如果是内部定义的接口，书写的时候你要每个层都提供一份声明。

最近在想提供一个东西做为隔离层，先看图。



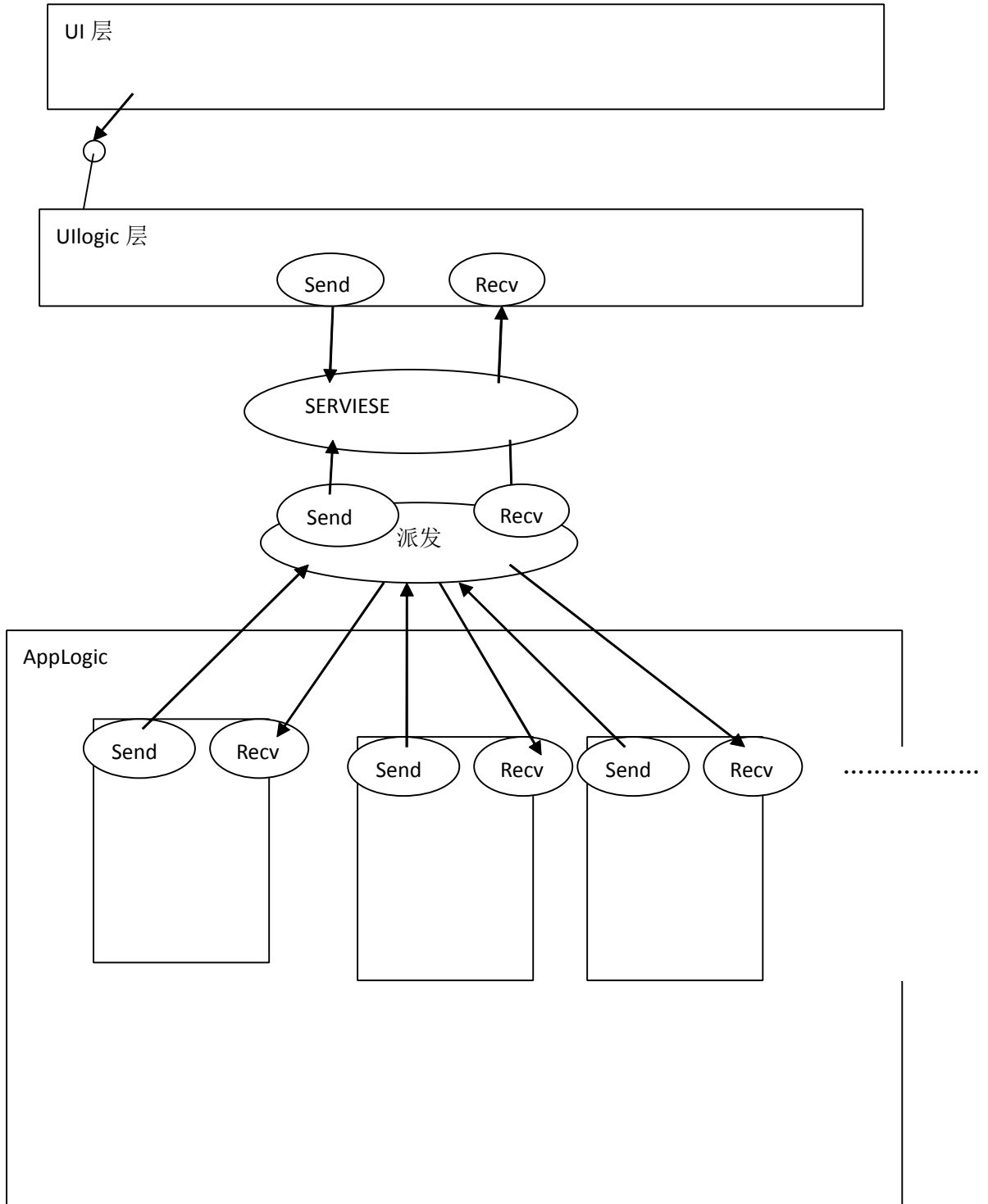
Applogic 和 uilogic 就只有数据间的通讯了。采用异步的消息方式，不管你是多线程还是单线程。模块之间耦合彻底消失。有人会问你这个 service 用啥。

用 Zeromq 吧，这是我机器上面测试的速度

<http://www.cppblog.com/cdy20/archive/2012/04/01/169791.html> 单线程一百 w 条 120s，那个快啊，如果作为本地的派发器的服务端足够了。

通信数据如何处理呢？zeromq 采用的是字符串发送，我们可以配合 google 的 protobuf 定义解析协议数据。这个东西可以压缩，解压缩协议非常方便，配上这个高速度的 mq，那简直棒极了。

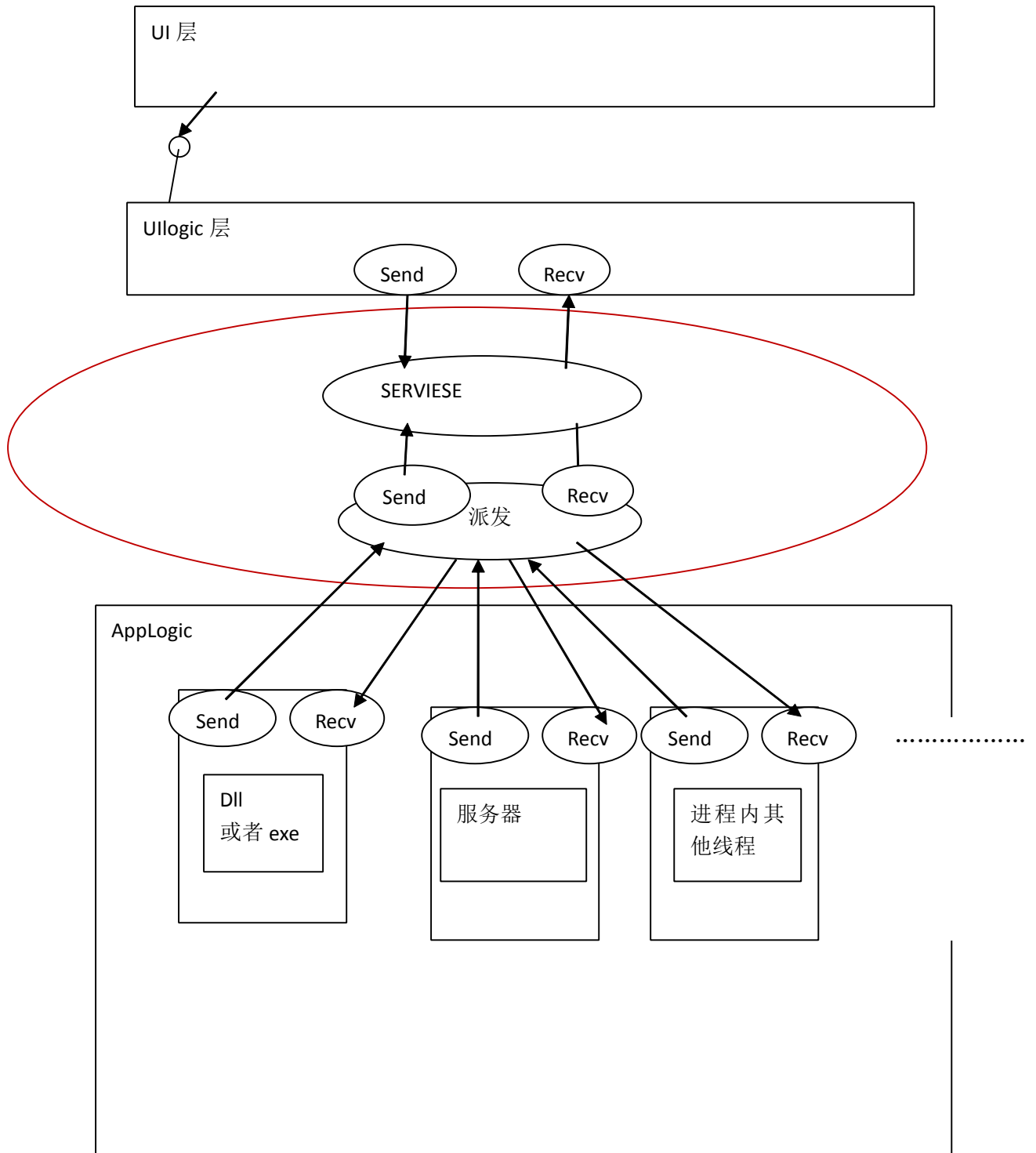
Zeromq 作为一个可以内核间、跨进程、跨机器通信的消息队列，不免又为我们提供了更多的设计的想象力。这么一个中间的消息的服务器，我们又可以把 applogic 切割，分成多个独立模块，如图。



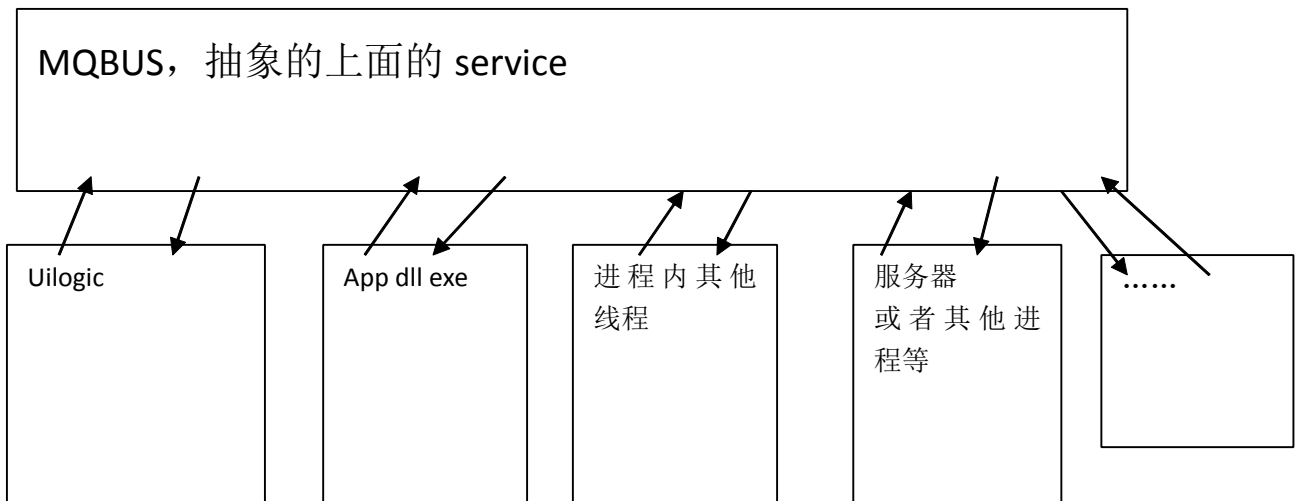
如上图，我们可以不断拓展 applogic 的模块，我们可以认为这些模块是独立的 动态链接库，独立的进程都可以，或者是独立线程，一切通信 进出都是通过发送 send，接收 recv。这个东西我现在遇到的工程项目组就有这个需求，他们要求是能够并发开发，甚至两个人统

一个模块不四个分支同时进行。一个强大的消息框架，让我的模块可以成为任何一种定义。

上面的图接口抽象的图如下。



红色区域代表消息的 bus。



这样程序就可以任意扩展了。任意切分了。

有错误和改进方法，[请发到 caidongyun19@qq.com](mailto:caidongyun19@qq.com). 谢谢阅读。
共享促进技术发展。蔡东赞。www.lomox.hk